Midterm examination

Disclaimer:
This is an open books exam. That means you are allowed to use any resources on your computer or the internet. **Exception to this** is any communication to others. This includes humans (chats, asking questions in forums (such as Stackoverflow, reading is allowed of course), shared documents etc.) as well as machines (As excited as we are about ChatGPT and similar technologies, we require that any responses submitted by you be generated exclusively by *your own biological neural network*). Please remember to close every messenger **completely**.

Multiple Choice Tasks:
In multiple choice tasks you will have an option to not answer. Make use of this if you don't know the answer instead of guessing, because wrong answers **subtract points** from your total points for that task.
Example 1: 3 correct answers, 2 wrong answers out of 5: total points: 1
Example 2: 2 correct answers, 3 wrong answers out of 5: total points 0

Beginn: 21.12.2023, 16:50
Ende: 10.12.2024, 19:00

Kurs: Implementing ANNs with TensorFlow (8.3304)
Semester: WiSe 2023/24
Lehrende: Prof. Dr. Elia Bruni, Serwan Jassim

Name: _____

1. Examination Code

0 Punkte

Please submit your examination code below

2. Data representation

4 Punkte

Different kinds of input data require different procedures. Fill in the gaps in the text below!

- Assume that all following examples of data are meant to serve as inputs to a neural network.
- Any answers regarding the **typical shapes** of such data must include the (mini-)batch size, denoted as **batch_size**.

- The most straightforward case is probably that of simple data vectors. Let's imagine a dataset of houses in which each house is described by a separate vector.
  - Let our vectors be three-dimensional and let the value of each dimension signify (1) the house's prize in Euros, (2) the house's height in meters, and (3) the year the house was built. Here are some examples:
    - [2.300.000 (€) ; 4,5 (meters); 1973 (a.d.)]
    - [4.500.000 (€) ; 6.3 (meters) ; 2012 (a.d.)]
  - By default, neural networks in TensorFlow expect inputs of the shape (batch_size, input_size).
  - Assuming that all values have already been cast to an appropriate datatype, there is still one major issue left to resolve before we are done preprocessing the data. Looking at the values and considering how we expect them to change throughout the dataset, we find that it is necessary to _____ normalize the values such that they are centered around $\mu =$_____ zero and approximate a std of $\sigma =$ _____ one
- We could also be working with image data. Let's go with cifar10 (you can click this link!) data:
  - Using mini-batches, we expect (as we generally do with image data in TensorFlow) an input shape of: _____ batch_size, height, width, channels
  - cifar10 is provided as the uint8 datatype. This means that each of the red, green and blue data channels is saved as an integer value in the range (if you are off by one you still receive the point!) _____ 0-255 .
  - Before μ, σ can be adjusted as stated above, one also has to change the datatype from uint8 to _____ float32 (the general name of the datatype is fine, we do not require the specific name used by TensorFlow)
- Sometimes we have to work with sequence or time-series data. Let's assume that we have a data vector describing the weather in Osnabrück on a daily basis, e.g. including temperature, precipitation, and hours of sunlight, i.e. for each day we get a data vector of length input_size.
  - Using this data in mini-batches, we expect (as we generally do with simple time-series data in TensorFlow) an input_shape of: _____ batch_size, sequence_length, input_size
  - Assuming a tf.data.Dataset (this is a link and can be clicked!) is used to prepare this time series data and we have sequences with very different sequence lengths: To create minibatches (with appropriate fill-in values where needed) one can use the respective _____ padded_batch function

Richtige Antworten hervorgehoben .

3. tf.data.Dataset steps for preparing the dataset

3 Punkte

While using tf.data.Dataset (this is a link, click it!), one typically invokes the respective .batch() , .shuffle() und .prefetch() functions.

- In which order should these typically be used? (1.5 pts)
- Why should they be used in that order? (1.5 pts)

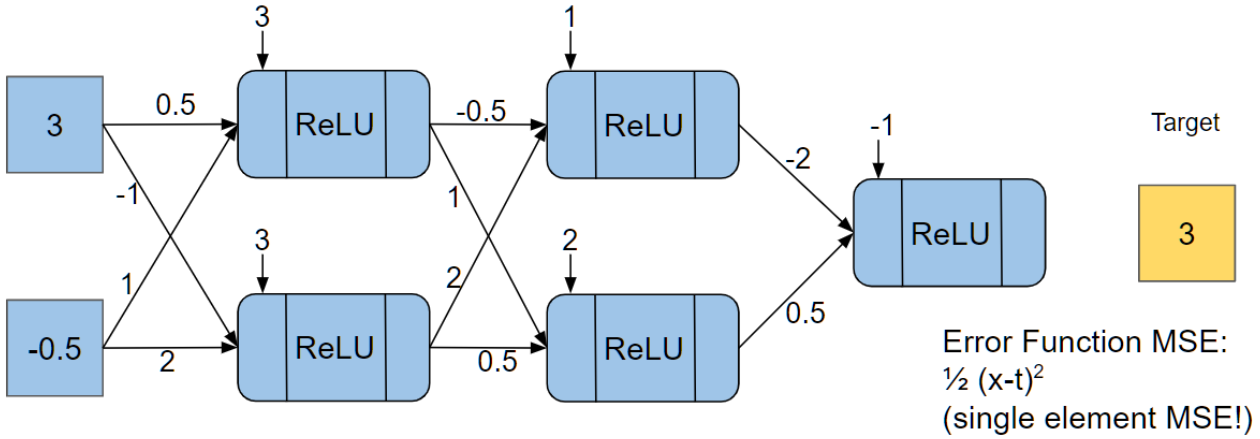
1.
First:
Second:
Third:

2.

4. MLP forward step

11 Punkte

Calculate the forward step for the MLP provided in the graphic below:



$x_1 = 3$
$x_2 = -0.5$

$w^{(1)}_{11} = 0.5$
$w^{(1)}_{12} = -1$
$w^{(1)}_{21} = 1$
$w^{(1)}_{22} = 2$
$b^{(1)}_1 = 3$
$b^{(1)}_2 = 3$

$w^{(2)}_{11} = -0.5$
$w^{(2)}_{12} = 1$
$w^{(2)}_{21} = 2$
$w^{(2)}_{22} = 0.5$
$b^{(2)}_1 = 1$
$b^{(2)}_2 = 2$

$w^{(3)}_{11} = -2$
$w^{(3)}_{21} = 0.5$
$b^{(3)}_1 = -1$

$t = 3$

Dateien zur Aufgabe:

- forward3.png

One example: W^2_{21} is the weight connecting the second neuron in the first layer and the first neuron in the second layer (superscript: Layer, subscript: neurons).

First Layer:

- $d_1^{(1)}$ _____ 4 (1 point)
- $d_2^{(1)}$ _____ -1 (1 point)
- $a_1^{(1)}$ _____ 4 (0.5 points)
- $a_2^{(1)}$ _____ 0 (0.5 points)

Second Layer:

- $d_1^{(2)}$ _____ -1 (1 point)
- $d_2^{(2)}$ _____ 6 (1 point)
- $a_1^{(2)}$ _____ 0 (0.5 points)
- $a_2^{(2)}$ _____ 6 (0.5 points)

Third Layer:

- $d_1^{(3)}$ _____ 2 (1 point)
- $a_1^{(3)}$ _____ 2 (0.5 points)
- Loss: _____ 0.5 (0.5 points)

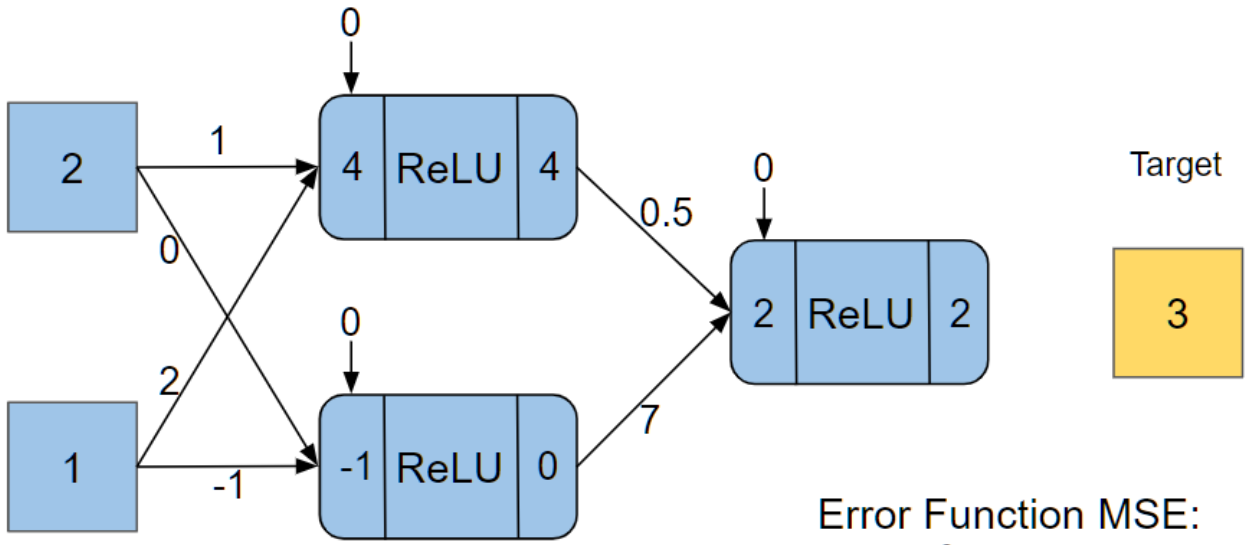
Richtige Antworten hervorgehoben.

5. Backpropagation

12 Punkte

Calculate the derivates as denoted below for the network in the following graphic:

Note: Below the graphic, for layer 2, the second time it says $w_{11}^{(2)}$, the correct subscript is 12, not 11, (value 7)



$x_1 = 2$
$x_2 = 1$

$w^{(1)}_{11} = 1$
$w^{(1)}_{12} = 0$
$w^{(1)}_{21} = 2$
$w^{(1)}_{22} = -1$
$b^{(1)}_{1} = 0$
$b^{(1)}_{2} = 0$

$w^{(2)}_{11} = 0.5$
$w^{(2)}_{11} = 7$
$b^{(2)}_{1} = 0$

$t = 3$

One example: W^2_{21} is the weight connecting the second neuron in the first layer and the first neuron in the second layer (superscript: Layer, subscript: neurons).

Second Layer:
Activation

- $\frac{dL}{da_1^{(2)}}$ = _____ -1 (1 point)

Net-Input

- $\frac{da_1^{(2)}}{dd_1^{(2)}}$ = _____ 1 (0.5 points)

- $\frac{dL}{dd_1^{(2)}}$ = _____ -1 (0.5 points)

Weights

- $\frac{dd_1^{(2)}}{dw_{11}^{(2)}}$ = _____ 4 (0.5 points)
- $\frac{dL}{dw_{11}^{(2)}}$ = _____ -4 (0.5 points)

- $\frac{dd_1^{(2)}}{dw_{21}^{(2)}}$ = _____ 0 (0.5 points)
- $\frac{dL}{dw_{21}^{(2)}}$ = _____ 0 (0.5 points)

First Layer:

Activation

- $\frac{dd_1^{(2)}}{da_1^{(1)}}$ = _____ 0.5 (0.5 points)
- $\frac{dL}{da_1^{(1)}}$ = _____ -0.5 (0.5 points)

- $\frac{dd_1^{(2)}}{da_2^{(1)}}$ = _____ 7 (0.5 points)
- $\frac{dL}{da_2^{(1)}}$ = _____ -7 (0.5 points)

Net Input

- $\frac{da_1^{(1)}}{dd_1^{(1)}}$ = _____ 1 (0.5 points)
- $\frac{dL}{dd_1^{(1)}}$ = _____ -0.5 (0.5 points)

- $\frac{da_2^{(1)}}{dd_2^{(1)}}$ = _____ 0 (0.5 points)
- $\frac{dL}{dd_2^{(1)}}$ = _____ 0 (0.5 points)

Weights

- $\frac{dd_1^{(1)}}{dw_{11}^{(1)}}$ = _____ 2 (0.5 points)
- $\frac{dL}{dw_{11}^{(1)}}$ = _____ -1 (0.5 points)

- $\frac{dd_1^{(1)}}{dw_{21}^{(1)}}$ = _____ 1 (0.5 points)
- $\frac{dL}{dw_{21}^{(1)}}$ = _____ -0.5 (0.5 points)

- $\frac{dd_2^{(1)}}{dw_{12}^{(1)}}$ = _____ 2 (0.5 points)
- $\frac{dL}{dw_{12}^{(1)}}$ = _____ 0 (0.5 points)

- $\frac{dd_2^{(1)}}{dw_{22}^{(1)}}$ = _____ 1 (0.5 points)

- $\frac{dL}{dw_{22}^{(1)}}$ = _____ 0 (0.5 points)

Richtige Antworten hervorgehoben.

6. TensorFlow

5 Punkte

The GradientTape is a function that makes the forward pass more efficient	○ True	○ False	○ keine Antwort
One reason to use a tf.data pipeline is that it makes it easy to efficiently load data without having to load all data into memory (RAM/VRAM) at once.	○ True	○ False	○ keine Antwort
The classes tf.keras.layers.Layer and tf.keras.Model are built on top of the tf.Module class	○ True	○ False	○ keine Antwort
The tf.Module class can not contain any variables.	○ True	○ False	○ keine Antwort
A tensor of shape (32,16,16,3) is said to be a rank 4 tensor.	○ True	○ False	○ keine Antwort

Richtige Antworten hervorgehoben.

7. Training Neural Networks

5 Punkte

Larger minibatches approximate the full batch gradient better	○ True	○ False	○ keine Antwort
It is always a good idea (i.e. there are no downsides to it) to use as large of a minibatch-size as possible	○ True	○ False	○ keine Antwort
A large batch size and a small learning rate can help escape local optima.	○ True	○ False	○ keine Antwort
Categorical Cross-Entropy is always a better loss function to use than Mean-Squared-Error	○ True	○ False	○ keine Antwort
For classification of cifar10, which contains 10 different classes, one should use the Binary CrossEntropy loss	○ True	○ False	○ keine Antwort

Richtige Antworten hervorgehoben.

8. Optimization I

4 Punkte

Adam is often seen as the default optimizer in Deep Learning. How does it address some of the following issues encountered in gradient descent:

1. The loss surface is very smooth and flat along the dimension of weight w_1, thus requiring large updates for efficient training. For weight w_2 the loss surface is highly curved with a sharp increase and decrease as w_2 is changed, thus requiring small updates. How does Adam account for different required update sizes for the different weights? (1 point)

2. Weight w_3 is updated with gradients showcasing iteratively alternating signs. This implies it skips over a local optimum repeatedly. How is this accounted for in adam?

3. Weight w_4 is updated with very small gradients (i.e. partial derivative), but each of them has the same sign. This implies it is stuck on a somewhat flat but consistent part of the loss surface. How is this addressed by adam?

1.

2.

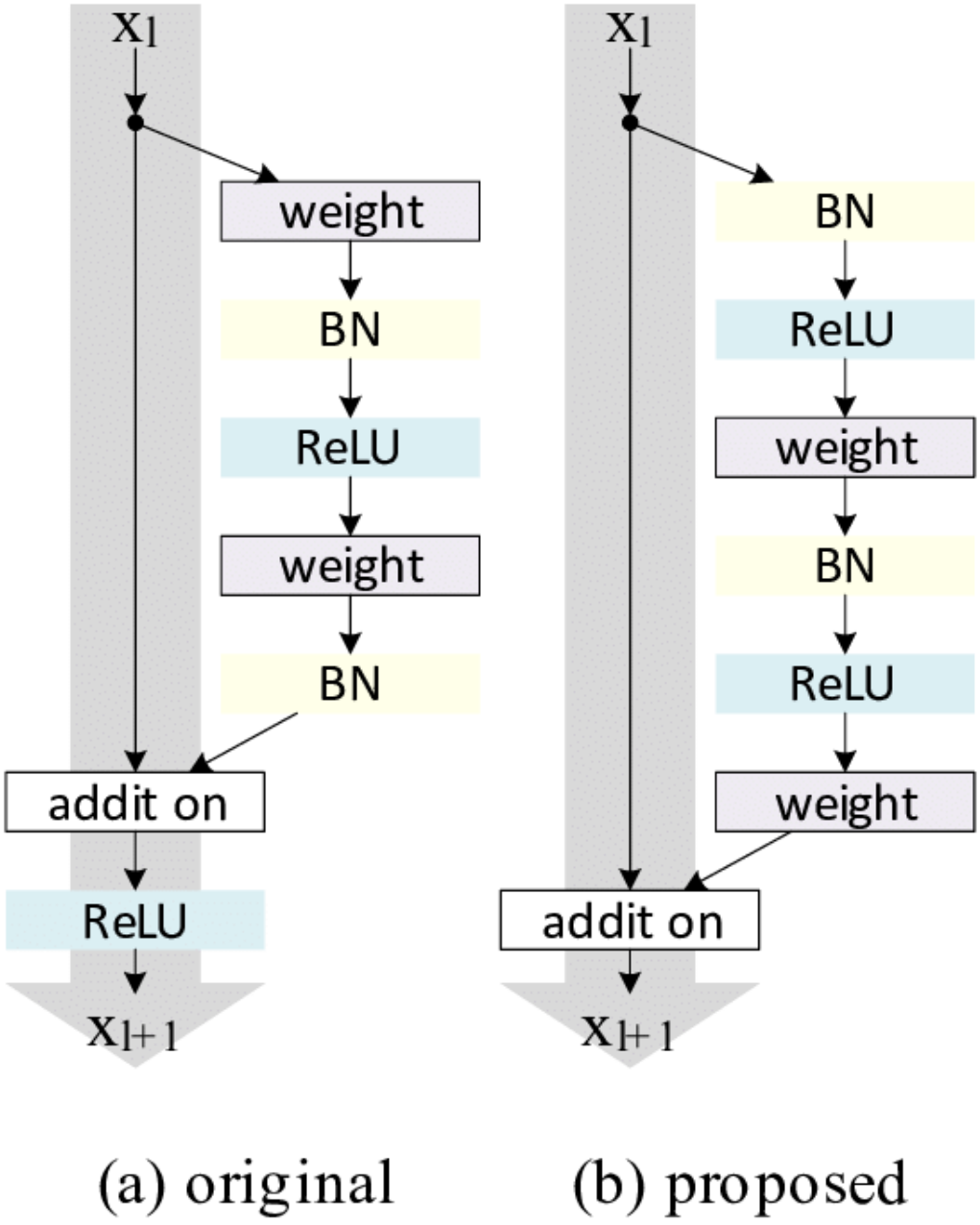
3.

9. TensorFlow and Resnet

4 Punkte

You are tasked with implementing the attached Resnet Layer (see file below). Specifically, you have to implement the forward pass, we already provide the class declaration and initialization:

FOCUS ON THE LEFT version from the two architectures provided below, you can also find the picture in the attached file again!



- You can expect the input having the same depth as the convolutions (the num_filters argument below)
- Notice this version of the Resnet Layer is slightly different from what we have seen in the lecture before!
- Make sure you follow the provided architecture closely!
- Use 4 spaces for indentation

```
class ResnetLayer(tf.keras.Model):
    def __init__(self, kernel_size, num_filters):
        super().__init__()
```

```
        self.conv1 = tf.keras.layers.Conv2D(num_filters, kernel_size, padding='same')
        self.bn1 = tf.keras.layers.BatchNormalization()
        self.relu1 = tf.keras.layers.ReLU() #using this layer is optional, you can also use
the respective tf.nn.relu() function

        self.conv2 = tf.keras.layers.Conv2D(num_filters, kernel_size, padding='same')
        self.bn2 = tf.keras.layers.BatchNormalization()
        self.relu2 = tf.keras.layers.ReLU() #using this layer is optional, you can also use
the respective tf.nn.relu() function

    def call(self, x, training=False):
        #TODO!!!
```

Dateien zur Aufgabe:

- resnet2.png

def call(x, training=False):

10. Overfitting

5 Punkte

Your artificial neural network may be overfitting!

1. How can you detect this using the graphs of loss (and accuracy) for training and validation data? (1 point)
2. Name four different approaches for how you could tackle this. Explain each in one sentence! (4 points)

1. How can you detect this using the graphs of loss (and accuracy) for training and validation data? (1 point)

2. Name four different approaches for how you cold tackle this. Explain each in one sentence!
2.1.
2.2.
2.3.
2.4.

11. Vanishing / Exploding Gradients

4 Punkte

Both the ResNet and LSTM architectures have been introduced to tackle vanishing gradients.

1. Why does each of them encounter this phenomenon, and how is this different between the two architectures (2 points)

2. What shared approach are both architectures using to tackle this problem? Pinpoint the math operation which both (among other ideas) rely on for this and explain in one sentence why it helps against vanishing gradients. (2 points)

1. Why does each of them encounter this phenomenon, and how is this different between the two architectures (2 points)

2. What shared approach are both architectures using to tackle this problem? Pinpoint the math operation which both (among other ideas) rely on for this and explain in one sentence why it helps against vanishing gradients. (2 points)

12. Calculating number of weights in different layers

4 Punkte

In the following, you are tasked with calculating the number of trainable weights (including all bias weights!) in the typical default configurations of the respective layers:

1. A LSTM with 20 units (i.e. size of cell-state = size of hidden-state = size of every timestep output = 20) is used with 10-dimensional input vectors, trained with TBPTT of sequence length of 70. It has a total of _____ 2480 trainable parameters (calculation: _____ 4*((10+20)*20+20)) (2 points)

2. A convolutional layer is applied to images of size 32x32 (with padding = 'valid'). The images are RGB images, i.e. they have three color-channels. The convolutional layer uses 3x3 filter-size (=kernel-size) and uses 14 filters and no bias. In total this convolution uses _____ 378 trainable parameters (calculation: _____ 3*(3*3)*14) (2 points)

Richtige Antworten hervorgehoben .

13. RNNs

3 Punkte

You want to tackle the following two tasks using RNNs (e.g. an LSTM):

1. Given all preceding words in a text, predict the next word! (Input: word sequence, Output: next word index)
2. Given a series of images (i.e. a video clip), predict a natural language text description of the clip (Input: video sequence, Output: word sequence)

Excluding considerations about the different data-forms (i.e. video, text/word), what is one crucial difference between these two sequence modeling tasks and how should this influence the respective network designs? (3 points)

14. Example Tasks

6 Punkte

In the following you can find two tasks to solve with a neural network model. Assume you have access to appropriate training data. For each of the two problems, describe...

1. The data format (shape, including batch size) and the model architecture you would use (2 points)
2. The output shape (including batch size) and output activation function you would use (0.5 points)
3. The loss function to train the model (0.5 points)

Problem I:
You have access to daily images of satellite images (RGB) of a country, data is recorded as 28 sequential days of such data. For each country you have such data samples multiple times each year, from the last 40 years. Additionally, you have access to a mask, showing where on each image the country in question is located (you may ignore this mask for the purpose of keeping your model simple in the following). You are tasked with predicting the amount of CO2 released in the respective country in the 28 days in each data-sample.

Problem II:
You are tasked with predicting the selling price of used cars. For each car you have an RGB image and some additional important information (year it was built, milage, etc). To predict the selling price you want to incorporate both the image data and the additional features in your model!

Problem I (CO2 prediction):
1.
Data shape:

Architecture:

2.
Output shape:
Output activation function:

3.
Loss function:

Problem II (car prices):

1.
Data shape:

Architecture:

2.
Output shape:
Output activation function:

3.
Loss function:

15. Bonus: Autoencoders

0 Punkte

This is a Bonus Task. You can achieve 100% of the score in this examination without answering this question at all. However you can receive up to two Bonus points for answering it successfully!

1. What is the conceptual difference between a basic Autoencoder and a Variational Autoencoder? (1 BONUS point)
2. Why is the reparameterization trick needed to train VAEs? (1 BONUS point)

16. BONUS: Read a paper!

0 Punkte

This is a bonus task. You can achieve 100% of the score in the examination without answering this question at all. Specifically, this question asks you to spend ~40 minutes just for achieving 1 bonus point, so we recommend you only work on this once you are done with everything else! You may see it as a possible reward for finishing the exam quickly.

Check out the paper '', which you can find here on arxiv: https://arxiv.org/pdf/1803.03635.pdf .
In one or two sentences each, answer the following:

1. What is the core idea of the Lottery ticket hypothesis? (1 BONUS point)
2. How are winning tickets found in the paper? (1 BONUS point)